

**METHOD AND SYSTEM FOR AGGREGATING INTERFACE ADDRESSES****BACKGROUND OF THE INVENTION**

## 5      1.    Technical Field:

10      The present invention relates in general to computer based networks and in particular to efficient destination address resolution at a node of a computer based network. Still more particularly, the present invention relates to a method and system for efficient destination address resolution by aggregating multiple interface addresses at a node of a network into a single representative maskaddress.

## 15      2.    Description of the Related Art:

20      In communications technology, an address is a unique designation for the location of data or the identity of an intelligent device. Multiple devices on a single communication line must have unique addresses to allow each to respond to its own messages and receive messages from other devices, particularly on more complex or widespread networks.

25      One of the more versatile and widespread communication networks is the Internet. The Internet is comprised primarily of multiple servers interconnected to each other. Each server has a number of connected clients which communicate with clients or remote servers utilizing  
30      Internet Protocol (IP) addressing. Transmitting destination specific data, such as email, represents one of the more

common uses of the Internet, as more and more users are appreciating the benefits of almost instantaneous transfer of information.

5           Currently, it is possible for a single host (such as a web server), to have lots of addresses associated with it. This is referred to as application hosting or web hosting, wherein a single machine may represent thousands of addresses. Occasionally, a node on a network is also  
10           designed to support (or host) multiple clients, with each client having a unique identifier or IP address.

Communication on the links are via packets (e.g., IP packets in a web based network and ethernet packet on a  
15           LAN). A communications packet is a group of bits, which includes information bits and overhead bits, and are transmitted as a complete packet on a packet switched network. The overhead bits include a destination address. When a packet is issued unto the network and arrives at a  
20           host, which has multiple addresses, the packet's destination address is compared against each of the multiple addresses in a sequential manner. This individual comparison is time consuming and results in a loss of efficiency for network packet transfer, particularly when multiple packets are  
25           routed simultaneously to and appear at the host at the same time.

Current IP address are typically 32 bits long. For example, IP address 199.72.6.100 is represented as 11000111  
30           01001000 00000110 01100100. In the 32 bit Internet Protocol (IP) addressing scheme, several billion addresses are possible and a node (or virtual server) is not limited, except for hardware constraints, as to the number of

addresses it may support. Some nodes may thus support hundreds of clients each having unique IP addresses. Attempting to locate an address by sequentially checking each address (i.e., all 32 bits) against the packet destination address may involve searching through hundreds or thousands of addresses hosted at the particular node.

For example, in a current systems, if a node supports IP addresses 192.35.45.4, 192.35.45.5, 192.35.45.6, and 192.35.45.7, the packet address is compared to each address sequentially. The comparison is completed at the host-network interface to determine if the packet is addressed to one of the addresses on the host and should be accepted.

The present invention recognizes that it would be desirable to provide a method and system for reducing the processing time for packets destination address resolution on a network. A system which minimized the time required for completing a comparison of a packet's destination address with addresses on a node (or virtual server) by reducing the actual number of comparisons and number of bits required to be compared would be a welcomed improvement. The present invention provides these and other improvements which are heretofore not previously provided.

## SUMMARY OF THE INVENTION

5 The present invention discloses a system for  
effectively reducing address resolution of a packet on a  
network by representing multiple addresses on a node of the  
network by a single maskaddress. An aggregation utility is  
utilized to aggregate the multiple addresses from an address  
space (i.e., a number of addresses having a common initial  
10 bits) of sequential addresses. A pre-selected aggregation  
percent, representing the percentage of actual addresses  
from the space which is present on the node, is utilized in  
the evaluation of the maskaddress. The maskaddress  
comprises of two variables. The first variable represents  
the size of the space and the common initial bits. The  
15 second variable provides information on the number of  
relevant bits to include in the maskaddress and the length  
of the space to point out the actual ones of the addresses,  
which are present in the space.

20 In a preferred embodiment, the system further comprises  
a negative-address utility by which those addresses which  
are not present within the space are identified. When a  
network packet arrives at the node, the destination address  
is compared against both the maskaddress and the negative  
25 address(es) to determine if to accept or refuse the packet.

The above as well as additional objects, features, and  
advantages of the present invention will become apparent in  
the following detailed written description.

**BRIEF DESCRIPTION OF THE DRAWINGS**

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself however, as well as a preferred mode of use, further objects and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

**Figure 1** depicts a data processing system, in which a preferred embodiment of the present invention may be implemented;

**Figure 2** is a high-level block diagram of an extended wide area network with interconnected Internet and local area network (LAN) in accordance with a preferred embodiment of the present invention;

**Figure 3** illustrates a high level logical flowchart of the process of 75% aggregating and negative addressing in accordance with a preferred embodiment of the present invention; and

**Figure 4** is a high level logical flow chart of the process of analyzing whether to accept or refuse a packet based on the aggregated and negative addressing scheme in accordance with one preferred implementation of the present invention.

**DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT**

The present invention is directed to a method for aggregating interface addresses within a computer based network. A computer system on which a preferred embodiment of the invention and a corresponding wide area network (WAN) are illustrated in **Figure 1** and **Figure 2**, respectively. The invention may, however, be implemented in other types of data processing systems and networks so, while the present invention may be described with references to the provided figures, these references should not be construed in a limiting sense. For the purposes of this invention, the term client is utilized to refer to both the hardware component which is connected to a network server and the software applications stored in memory and being run on the hardware component. Each client has a corresponding IP address. Additionally, the terms node, host or virtual server as utilized herein refer to a point on the network which provides a service, specifically that of hosting one or more clients or client addresses.

With reference now to the figures and in particular with reference to **Figure 1**, a data processing system that may be utilized as a standalone data processing system or one of the client terminals or servers on a network is presented. Data processing system **20** comprises a Central Processing Unit (CPU) housed in a system unit **22**. System unit **22** also provides connections for various hardware components including disk drives **40**, and memory devices (not shown). Stored within memory devices are the operating system (OS) and software applications by which many of the

processes of the invention are implemented as will become clear later. Several peripheral input/output devices are connected to the CPU. These input/output devices include keyboard 82, mouse 84, printer 94, CD ROM 78, and display monitor 30. Display monitor 30 provides a graphical user interface (GUI) which allows a user to view and interact with software applications 26 stored in system memory or provided via a network, by displaying icons or other selectable mechanisms with which the user may interact.

Also coupled to CPU are various external devices, including modem 92, and network adapter 90, both utilized for connecting data processing system 20 to other systems and/or networks, as is illustrated in **Figure 2**. CD ROM 78, modem 92, and network adapter 90 are depicted as external components; however those skilled in the arts are familiar with the various structures of data processing system architecture and understand that these components may be housed inside of system unit 22.

Modem 92 is a communication device that enables a computer to transmit information over standard telephone lines or wireless connections such as cellular. Modem 92 converts digital computer signals to interlock signals suitable for communications over these telephone media. Modem 92 can be utilized to connect data processing system 20 to a server via remote access protocols. Modem 92 may also provide a connection to other sources, such as an electronic bulletin board (BBS) or the World Wide Web.

Network adapter 90 may be used to connect data processing system 20 to a Local Area Network (LAN) as depicted within **Figure 2**. LAN may consist of a single server or a number of servers and provides network clients with a means of communicating and transferring software and information electronically within the network environment.

Referring now to **Figure 2**, a high-level block diagram of an extended wide area network (WAN) is illustrated in accordance with a preferred embodiment of the present invention. The preferred embodiment of the present invention is concerned with the aggregation and subsequent representation of multiple IP addresses located a node as a single maskaddress so that processing time for evaluating a packet on the network is reduced. IP addresses are represented with (1) a header V,W,X, Y, etc., which identifies the addresses with a common physical address/layer, and (2) a 3-bit logical extensions (000 -111) of the physical address, identifying the specific client. Thus V001 may represent IP address 192.53.45.1 and V110 may represent IP address 192.53.45.6., etc. Each node hosts several IP addresses (e.g., 192.53.45.4, 192.53.45.7, and 192.53.45.5). The complexity of current networks thus results in multiple physical layers, which have a single header to accommodate each client supported by that header.

WAN 201 includes the Internet (Web) 211 as well as a local area network (LAN) 202. Internet 211 contains a plurality of servers 213 which are connected within a web like structure scattered within a geographically obscure area depicted herein as a cloud 212. The server may be an



ISP server 213, having many users 217A and associated IP addresses 217B at a single node. Information server 214A has no direct client connections, but also has an associated web or IP address 214B by which information server 214A may be accessed by remote clients. The preferred embodiment of the invention is concerned with the multiple-address nodes 219A and 209A, which host multiple IP addresses 219B and 209B.

LAN is comprised of a plurality of client systems 205A coupled to central servers 203 via an ethernet, for example. Central servers 203A and client systems 205A have affiliated IP addresses 203B and 205B, respectively. The invention finds applicability with a multiple-address node of both a LAN system and the Internet as described herein but may be extended to any network structure in which multiple users or clients have associated IP address, which are hosted at a single node in the network.

In the preferred embodiment, Each of the nodes on WAN 201 are computer systems or coupled to computer systems, which have a database, OS, and address-aggregation and interfacing software utility. The invention is implemented utilizing one or more of the following utilities: (1) an aggregation utility; (2) a negative address (anti-address) utility; and (3) an address comparison utility. Although the first two utilities may be utilized independent of each other, the preferred embodiment of the invention utilizes both the aggregation and negative-address utilities together to improve the efficiency of the process and substantially maximize the accuracy of the resulting mask address(es).

In the illustrative embodiments, a 75% aggregation method is utilized. This means that the aggregating utility only aggregates addresses within a space when at least 75% of the possible sequential addresses are present on the node. The various processes of the invention are completed as software code resident on the node's computer system or other computer system connected to the node.

The invention operates to find the smallest address space which encompasses the physical address that includes 75% of the logical addresses. In application of the invention to 32 bit addresses, the first 28 bits may be the same (i.e., encompassing the physical address), with the final four bits being utilized to provide the sixteen (16) individual IP addresses which may be hosted at the node. The mask address for the 16 IP addresses, using a 75% aggregation method, would include only the first 28 bits. If only 75% of the bottom eight (i.e., addresses 8-15) address were present at the node, then the mask address would be 29 bits. The 29 bits would comprise the first 28 bits plus an additional bit, "1," which is common to the bottom eight addresses.

The invention requires the aggregation utility to create at least one mask address to represents an entire space of addresses on a single node. In order to begin the aggregate comparison, the aggregation utility first determines which address bit is important. In a 32 bit IP addressing scheme, this involves finding the last bit(s) that are common to all the addresses for a node. Then, the bit is selected so that all the elements in the space have the same primary bit. For example, a set of 32 bit binary

addresses with a common first 29 bits would be represented by a mask address comprising the first 29 bits in addition to a second set of values, which indicate which of the eight (8) possible addresses (i.e., 000-111 binary, representing bits 30-32) are present in the space. Thus, the first function of the aggregating utility is to compare the addresses on the node to find commonalities in the initial address bits. Once a commonality is found for a set of the addresses, the address space can be determined and the other functional processes of the aggregation utility completed. In the preferred embodiment, the aggregation and negative address utility are activated when the node is first made to host more than one address or when the utility is loaded into the node computer system. Subsequently, whenever a change occurs at the node, such as addition or removal of a client or IP address, the utilities are triggered to complete the aggregating steps again. These functional processes occur independent of the network and the transmission of packets on the network.

Returning to the above example where a space of 8 possible addresses (i.e., 3 relevant bits) is defined, if the value of all of the first bits is 1 then the space is reduced to four (4) possible addresses (i.e., 4,5,6, and 7). Likewise, 0 as the value of the first bit reduces the space to the other four addresses (0,1,2, and 3). Thus if the incoming address has a first relevant bit value of 0, there is no match within the aggregate space of 4,5,6, and 7. For purposes of simplicity and to ensure clarity, the invention will hereinafter be described with reference to this 8 address space, and in particular to the subset of 4 address space (i.e., 4,5,6 and 7).

In the presented 3-bit example, the first bit is most relevant because a value of 1 is required for that bit for an address to be on the node. Any packet destination address with a value of 1 in that bit, would possibly find a match within the space. (Likewise, a 0 value in that bit would be a clear indication that the packet address is not on the node). During the actual comparison, only those bits of the address up to the first bit need to be checked against the first part of the mask address for a match. The other bits are thus eliminated from the network level comparison process. Also, only a single comparison of the destination address with the mask address is required. The invention thus provides a mechanism for optimally aggregating the addresses in a larger space to reduce packet comparison time, which increases overall network efficiency.

In the given example, once a packet appears at the node with a 1 in the position of the first bit, the packet is accepted and subjected to further checks against the other components of the maskaddress. The comparison utility completes a network-level check up to the first bit of the maskaddress, but completes a node-level check against the second and subsequent bit level. This node-level check continues until the packet finds the correct address or the address is determined to be not within the node or nonexistent. Thus, from the network level, only the initial comparison with the mask address is required. Once the destination address matches the maskaddress up to the first bit, all other comparisons occurs internally within the host.

The invention utilizes two variables within its aggregation utility to represent the space and addresses

found within the space. These variables and their definitions and/or functional characteristics are:

P: this represents the address space. For example, in the 3 bit address space,  $P = 8$  (i.e., 8 possible addresses in the space);

$P_k$  (also referred to herein as the prefix): this represents the particular part of the address space which is important.

$P_k$  contains two pieces of information, (1) the particular setting of bits, and (2) the number of bits which are important or which are in the prefix. For example, a 3 bit address space with the first bit value of 1 has a prefix of 100, where only the first bit is important. Thus the presented example is coded as:

$$P=100, P_k=100/1.$$

The actual addresses of relevance for the aggregate are 100-111 (4-7), each with a first bit value of 1. The shorter the prefix, the larger the number of relevant addresses that are present at the node and vice-versa. During the comparison of a packet destination address, if the destination address matches a space up to the prefix length, then the destination address is assumed to belong to that node pending further analysis.

A further example, coded  $P=100$ ,  $P_k=110/2$ , represents that although the space has 8 possible addresses, the number of relevant addresses in the space is 4, and the first two bits are relevant when comparing the maskaddress with a destination address. This translates to an aggregate, which represents address values 6 and 7.

The following implementation of the invention follows a 75% aggregation rule. Use of a 75% aggregation rule is not meant to be restrictive on the invention and was chosen solely for illustrative purposes. It is preferable to have as high a percent as possible. Thus, ideally a 100% rule would be followed, but this would be impractical in most networks.

The term 75% rule refers to the percentage of the addresses within a given sequential space which is required to be present on the node to be represented by a single mask address. The invention thus attempts to aggregate the largest space with 75% of the address covered. Addresses are typically assigned sequentially, though not necessarily to the same node. Thus when all address between 0-7 in the 3-bit addressing scheme are assigned, the 75% rule requires that at least 6 of the 8 addresses are located at the node to have a representative space of 8. If, for example, only addresses 4,5,and 7 are present on the node, then the largest representative space is actually 4, and so on. The invention thus first completes a binary search within the entire space to determine the largest prefix with the smallest prefix length having 75% of the addresses. The above example yields values of  $P=100$ , and  $P_k=100/1$ . However, because there are possibly other addresses in the total space not covered in  $P=100$ , other aggregating steps must be taken to cover these other addresses as well.

The process of obtaining the 75% aggregate address is an iterative process, wherein each iteration further reduces the space, increases the number of prefixes in that space by a factor of 2, and increases the number of bits in the length of  $P_k$  by 1.

Use of an aggregation process with less than a 100% address requirement causes a number of "holes" to appear where the number of addresses are uneven or the addresses are not present on the node sequentially, or the number of address is not a factor of 2. A second utility, the negative-address utility is implemented to account for (or fill) the holes. The various utilities may be nested within each other in or exist as separate functional components of the invention.

From an implementation standpoint, the negative address method is the reverse of the 75% method and actually collects information on which addresses are not present in the total space. In the general example, if only addresses 4, 5 and 7 are present in the space, then a negative prefix exists with a prefix length of 3 and a value of 6.

When a space is identified as not having all of the sequential addresses, the space is further divided to isolate the address not present. Thus prefixes with a prefix length of 2 are checked. The first subspace of 2 (100) with a 2 bit prefix length yields 4 and 5, which are 100% covered (i.e., no holes). The second subspace of 2 (110) with a 2 bit prefix length yields only 7, which is less than 75% covered. The algorithm then moves on to the next iteration level and checks the prefix length of 3 against the next level subspace (111). At this level, only 1 subspace (which represents 7) is greater than 25% uncovered. The negative address utility represents this as  $P_k = -111/3$ .

The combination of Positive and Negative address spaces ( $P_{k+} = 100/1$  and  $P_{k-} = 111/3$ ) completely defines the entire

set of addresses from within a particular space which is hosted on the node. The applicability of the present invention is extended to and works more efficiently within larger networks wherein, for example, the addresses are 8 bits long (or longer), representing 256 possible addresses within the total address space.

In yet another example, using a node having addresses 0,1,4,5,6, and 7. The addresses would aggregate to two addresses 000/2 and 100/1. Similarly, there may be multiple holes in an aggregate. So for example, a node with addresses 1,2,3,4,5, and 6 would aggregate to three addresses - one positive (000/0) and two negative (-000/3 and -111/3). When the order (i.e., number of bits) increases, the aggregates and holes become nested. A four bit address space, for instance, may present a node having addresses 3 to 15 (i.e., the entire space except addresses 0,1, and 2). This space may be represented by three aggregates (1) 0000/0 (the entire space because there are more than 75% of the addresses present), (2) -0000/2 (addresses 0,1,2, and 3 represents more than 75% negative), and (3) 0010/4. Thus, for each subspace that is identified as a potential aggregated or negative address, the invention continues to divide the space until it is completely described.

An example is now presented of a node, which has at least four virtual servers or clients represented by 32-bit IP addresses 192.35.45.4, 192.35.45.5, 192.35.45.6, and 192.35.45.7, etc. According to the preferred embodiment of the invention, the IP addresses are aggregated into a maskaddress with prefix information. The address that is then represented to the network is 192.35.45.4/30. This maskaddress indicates that the first 30 bits are relevant to



the comparison with an incoming packet destination address. The comparison utility makes a single comparison of the maskaddress with the incoming packet destination addresses up to the first 30 bits. Thus, in this example, the invention permits the replacement of the four individual address comparisons with a single, maskaddress comparison. One skilled in the art will appreciate that as the number of aggregated addresses increase, the average time saved per packet will also increase. Applicability of the invention to larger networks having greater numbers of addresses is preferred. By utilizing the aggregated shorthand (maskaddress), an incoming packet can be tested against multiple addresses simultaneously, reducing the per packet overhead of an IP network stack.

Turning now to **Figure 3**, there is presented a flow chart of the processes involved in aggregating addresses according to one preferred embodiment of the invention. Process steps as described are preferably completed by a software-based utility running on a data processing system within the network. The process begins at block 301. An initial binary search of the entire total space on the node is completed at block 303. This search is used to determine whether 75 percent of the addresses are present within the total space at block 305. If 75% is not found, then the total space is divided into sub-spaces at block 306. The number off sub-spaces doubles and the length of each prefix of a subspace increases by 1. This division of the space into smaller unit sub-spaces continues until the query at block 305 yields a positive result. Thus when 75% of the addresses within the space (or sub-space) is present, a second check is made to determine whether there are holes

within the space (i.e., missing addresses). If there are no holes within the space (100% present), then the space size and prefix are saved on the node computer at block 308 and utilized during execution of a packet response at block 317.

5 If, however, there are holes in the space, a negative addressing process is initiated at block 309. The space is further divided in block 311 as in block 306; however, the subsequent check made is to determine if there is less than 75% (greater than 25% missing) of the addresses in the space  
10 at block 313. This represents the condition of interest for the negative address check. When the space no longer has more than 25% of the addresses missing, the negative address values of the space and prefix are set at block 315. The process then proceeds with setting the positive values at  
15 block 308. Thus the process of dividing a space both in the aggregation and the negative address method, is an iterative one, which continues until all the addresses within (or not in) the space hosted on the node are accounted for. This may result in multiple mask addresses and multiple negative  
20 addresses representing particular sub-spaces within the larger space.

**Figure 4** illustrates the execution process of address resolution on a network once the aggregate addresses have  
25 been set. The process begins at block 417. A packet with a specific IP destination address is received at the node at block 419. The node's software logic compares the IP destination address with the mask address (or aggregate address) at block 421. The logic determines if a match  
30 exists between the two addresses at block 423. The process of comparing the addresses involves a bit by bit comparison

of the addresses beginning with the first or leftmost bit (also referred to as the most significant bit). The comparison continues until the respective bits fail to match or the number of bits in the maskaddress have been compared.

5 If a match does not exist, then the packet is refused at block 425 and the process ends at block 433. If, however, a match exists, then the destination address is compared with the negative address(es) at block 427 and a determination of a match is completed at block 429. If a match to the

10 negative address(es) exists, then the packet is refused at block 425. Otherwise, the node's comparison utility triggers the node's process for checking internal addresses and locates the specific IP destination address within the node. The packet is then forwarded to the corresponding

15 destination address at block 431. The process ends at block 433.

Those skilled in the arts understand that the above processes are presented as steps which are completed primary by program code. Different process steps may be provided and in a different order than those of **Figures 3 and 4**. The steps as presented in these figures are meant to illustrate only one possible representation of the invention and are not meant to be limiting on the invention.

25 It is important to note that while the present invention has been described in the context of a fully functional data processing system, those skilled in the art will appreciate that the mechanism of the present invention is capable of being distributed in the form of a computer readable medium of instructions in a variety of forms, and that the present invention applies equally, regardless of

30

the particular type of signal bearing media utilized to actually carry out the distribution. Examples of computer readable media include: nonvolatile, hard-coded type media such as Read Only Memories (ROMs) or Erasable, Electrically Programmable Read Only Memories (EEPROMs), recordable type media such as floppy disks, hard disk drives and CD-ROMs, and transmission type media such as digital and analog communication links.

5

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209

While the invention has been particularly shown and described with reference to a preferred embodiment, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention. Particularly, while the invention has been described with primary reference to Internet-based network systems, the invention finds applicability in LAN based networked systems, as presented in **Figure 2**.

In LAN systems, which utilize address resolution protocol (ARP), each node may support multiple client IP addresses and is connected to a router to determine its machine address stored within a mapping table. When the node supports multiple client IP addresses, it creates a mask address which is forwarded to the mapping table. When the router receives a data packet, it checks its mapping table to see if there is an entry for the IP address received. The router compares the destination address with the mask address and if there is a match, the router creates a packet (such as an Ethernet packet or Token Ring) and transmit the data over the LAN. If there is no match then the router sends an ARP broadcast message over the LAN. Utilizing the processes of the invention, the individual addresses within the mapping table is thus replaced with the maskaddress.